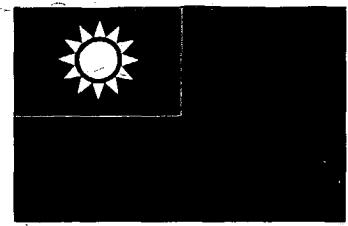


US-4285



# 中華民國經濟部智慧財產局

INTELLECTUAL PROPERTY OFFICE  
MINISTRY OF ECONOMIC AFFAIRS  
REPUBLIC OF CHINA

茲證明所附文件，係本局存檔中原申請案的副本，正確無訛，

其申請資料如下：

This is to certify that annexed is a true copy from the records of this office of the application as originally filed which is identified hereunder:

申請日：西元 2002 年 07 月 09 日  
Application Date

申請案號：091115380  
Application No.

申請人：蔚華科技股份有限公司、國立清華大學  
Applicant(s)

局長

Director General

蔡練生

發文日期：西元 2003 年 1 月 9 日  
Issue Date

發文字號：09220021560  
Serial No.

28673

申請日期	
案 號	
類 別	

A4

C4

(以上各欄由本局填註)

## 發明專利說明書 ~~新型~~

一、發明 名稱  <u>新型</u>	中 文	快閃記憶體之對角線測試方法
	英 文	
二、發明人 創作	姓 名	1. 邱 紹 國                  5. 吳 誠 文 2. 葉 人 傑 3. 鄭 國 良 4. 黃 稚 存
	國 籍	均 中 華 民 國
三、申請人	住、居所	1. 苗栗縣通霄鎮城南里11鄰134之1號 2. 台中縣大雅鄉神林南路173巷2弄12號4樓 3. 台南縣新營市太南里3鄰156號之1 4. 嘉義市大雅路二段386巷29號 5. 新竹市光復路二段101號
	姓 名 (名稱)	1. 蔚華科技股份有限公司 2. 國立清華大學
	國 籍	均 中 華 民 國
	住、居所 (事務所)	1. 新竹市復中里自由路69號6樓之1 2. 新竹市光復路二段101號
	代表人 姓 名	1. 陳 有 諒 2. 徐 遲 生

四、中文發明摘要(發明之名稱：快閃記憶體之對角線測試方法 )

本發明揭示一種快閃記憶體之對角線測試方法。該測試方法係將該快閃記憶體虛擬為數個正方形，並以由上至下和左至右的方向執行。每一個正方形具有一由左上至右下之 -45 度之第一對角線及一由左下至右上之 +45 度之第二對角線。本發明即利用程式化該第一對角線或第二對角線之元件，且之後讀取除該第一對角線或第二對角線以外之元件；或程式化除該第一對角線或第二對角線以外之元件，且之後讀取該第一對角線或第二對角線之元件的方式而快速有效檢測該快閃記憶體之干擾錯誤與一般記憶體之錯誤。

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

訂

線

英文發明摘要(發明之名稱： )

(由本局填寫)

承辦人代碼：
大類：
I P C 分類：

C6  
D6

本案已向：

國（地區）申請專利，申請日期： 案號： ，有 無主張優先權

本案在向中華民國提出申請前未曾向其他國家提出申請專利。

(請先閱讀背面之注意事項再填寫本頁各欄)

裝

訂

線

## 五、發明說明 ( 1 )

### 發明領域

本發明係關於一種快閃記憶體之測試方法，特別是關於一種快閃記憶體之對角線測試方法。

### 發明背景

快閃記憶體係一種可程式及可抹除記憶資料之非揮發型記憶體。在近年來快閃記憶體的需求量快速地增加，特別是應用於無線傳輸、自動化設備、資料儲存、機上盒 (set-top box) 和各種多媒體產品應用。快閃記憶體亦已大量地整合入包含各種邏輯電路和記憶體核心 (core) 的系統單晶片 (system on a chip, SOC) 內，特別是以電池為電源的可攜式產品。由於系統單晶片的設計多傾向於高密度、高容量及高腳位，因此測試的時間和難度便大幅地提高。如何快速且有效地降低測試的時間以提高產出的效率，便成為一項重要的課題。

一般而言，快閃記憶體之測試或診斷難度高於傳統之記憶體，特別是因同一行或同一列元件在讀寫時所產生的干擾問題 (disturbance problem)。IEEE 1005標準規範了快閃記憶體的干擾問題及其影響，例如因在高電壓操作下進行記憶體讀取或程式化 (programming) 而導致相鄰之記憶體元件非預期之電荷移轉等。和SRAM或DRAM之操作方式不同的是，快閃記憶體並非僅寫入單一的位元或字元；相反地，快閃記憶體可進行一整個區塊或晶片的抹除 (flash erase)，例如同時將邏輯1寫入整個區塊或晶片。

近年來，各種不同測試方法不斷被提出。其中在M. G.

## 五、發明說明 ( 2 )

Mohammad, K. K. Saluja, and A. Yap, "Testing Flash Memories", in Proc. 13th Int. Conf. VLSI Design, Jan. 2000, pp. 406-411之論文內揭示一種新的干擾錯誤和最佳演算法（以下簡稱為EF）。在M. G. Mohammad and K. K. Saluja, "Flash Memory Disturbances : Modeling and Test", Proc. IEEE VLSI Test Symp., Marina Del Rey, California, April. 2001, pp. 218-224之論文內，將快閃記憶體之干擾錯誤模型區分成數種耦合錯誤(coupling faults)，並提出一有效的March-like演算法（以下簡稱為Flash March）以偵測該耦合錯誤。在J. C. Yen, C. F. Wu, K. L. Cheng, Y. F. Chou, C. T. Huang, and C. W. Wu, "Flash Memory Built-in Self-test Using March-Like Algorithms", Proc. IEEE Int. Workshop on Electronic Design, Test, and Applications (DELT), Christchurch, Jan. 2002, pp. 137-141之論文內揭示一種位元導向和字元導向的March-like演算法（以下簡稱為March-FT），用以涵蓋IEEE 1005標準定義之所有干擾錯誤型態。

對快閃記憶體之操作而言，抹除操作之運算速度最慢，其次為程式化操作，而讀取操作的速度最快。然而習知之March-like演算法包含許多的寫入操作（包含抹除操作和程式化操作），其所花費之測試時間仍嫌過長，以致測試成本仍居高不下。如何設計一有效之測試方法，且在不降低錯誤涵蓋率(fault coverage)的情形下仍可有效地減少測試所需的時間，即成為現今一項非常重要之課題。

### 發明之簡要說明

## 五、發明說明（3）

本發明之主要目的係提供一快閃記憶體之對角線測試方法，在不降低錯誤涵蓋率的情形下，仍可有效地減少測試所需的時間。

本發明之第二目的係提供一快閃記憶體之測試方法，其不僅可應用於測試軟體之製作，亦可應用於內建自我測試電路之製作。

為達成上述目的，本發明之快閃記憶體之對角線測試方法包含步驟(a)至步驟(e)。在步驟(a)中，抹除整個快閃記憶體元件陣列。在步驟(b)中，程式化除第一對角線以外之元件。在步驟(c)中，讀取第一對角線之元件。在步驟(d)中，程式化第一對角線之元件。在步驟(e)中，讀取除第一對角線以外之元件。

本發明之快閃記憶體之對角線測試方法可另包含步驟(f)和步驟(g)。在步驟(f)中，程式化第二對角線之元件。在步驟(g)中，讀取第一對角線之元件。步驟(f)及(g)和步驟(a)至(e)沒有必然之執行順序關係，測試程式開發人員可依程式撰寫時之便利性而決定先執行步驟(f)及(g)或步驟(a)至(e)。

本發明之快閃記憶體之對角線測試方法可另包含步驟(h)至步驟(j)。在步驟(h)中，抹除該快閃記憶體元件陣列。在步驟(i)中，依相反於第一對角線之方向而程式化位於該快閃記憶體元件陣列之第一對角線之元件。在步驟(j)中，讀取該快閃記憶體元件陣列之除第一對角線以外之元件。步驟(h)至(j)和步驟(a)至(e)或步驟(f)及(g)沒有必然之

## 五、發明說明 ( 4 )

執行順序關係，測試程式開發人員可依程式撰寫時之便利性而決定先執行步驟(h)至(j)，或步驟(f)及(g)或步驟(a)至(e)。

在本發明之步驟(j)中，可使用一延伸型之演算法而使錯誤涵蓋率達到100%。而當使用延伸型之演算法時，步驟(f)則可予以省略。

在上述之測試方法中可將該快閃記憶體虛擬為數個正方形，並以由上至下和左至右的方向執行。每一個正方形具有一由左上至右下之-45度之第一對角線及一由左下至右上之+45度之第二對角線。本發明即利用程式化該第一對角線或第二對角線之元件，且之後讀取除該第一對角線或第二對角線以外之元件；或程式化除該第一對角線或第二對角線以外之元件，且之後讀取該第一對角線或第二對角線之元件的方式而檢測該快閃記憶體之干擾錯誤與一般記憶體錯誤模式(fault models)。

上述之第一對角線及第二對角線之執行方向-45度及+45度僅係一實施例，在實際應用中可依設計者之偏好而予以修正，而本發明亦未限定該第一對角線及第二對角線為任一特定之執行方向。

本發明之偵錯範圍除該快閃記憶體之干擾錯誤外，亦可涵蓋傳統之記憶體錯誤。

利用上述測試方法所形成之電路僅佔據很小之面積。對一般常使用的快閃記憶體而言，大約為2仟至3仟個閘數(gate count)。本發明之測試方法所花費之測試時間大約

裝  
訂  
線

## 五、發明說明(5)

較習知之March-like演算法降低42.69%。因此本發明之測試方法確能在不降低錯誤涵蓋率的情形下，仍可有效地減少測試所需的時間，實具明顯之新穎性及高度之進步性，且為利用自然法則之技術思想之高度創作。

### 圖式之簡單說明

本發明將依照後附圖式來說明，其中：

圖1(a)至1(c)係本發明之快閃記憶體之對角線劃分示意圖；

圖2係本發明之第一實施例之測試流程圖；

圖3(a)至3(g)係第一實施例之步驟分解圖；

圖4(a)至4(i)係本發明之第一實施例之另一分解表示圖；

圖5(a)至5(i)係本發明之第二實施例之測試步驟分解圖；

圖6(a)至6(i)係本發明之第三實施例之測試步驟分解圖；

圖7(a)至7(i)係本發明之第四實施例之測試步驟分解圖；

圖8(a)至8(i)係本發明之第五實施例之測試步驟分解圖；

圖9(a)至9(h)係本發明之第六實施例之測試步驟分解圖；

圖10(a)至10(h)係本發明之第七實施例之測試步驟分解圖；及

## 五、發明說明（6）

圖 11(a) 至 11(h) 係本發明之第八實施例之測試步驟分解圖。

### 較佳實施例說明

本發明之快閃記憶體之對角線測試方法主要係為解決快閃記憶體之干擾錯誤及一般記憶體之錯誤模式。由於快閃記憶體將高電壓應用於浮動閘記憶體之端點上，因而造成可能的干擾問題，而受干擾效應影響的元件通常位於被程式化或被讀取元件之同一位元線或同一字元線。一般而言，該干擾錯誤模式可分為：

1. 字元線程式化干擾（Word-line Program Disturbance，WPD），其發生原因在於程式化一元件後導致同一字元線之未程式化元件亦被程式化。
2. 字元線抹除干擾（Word-line Erase Disturbance，WED），其發生原因在於程式化一元件後導致同一字元線之已程式化元件被抹除。
3. 位元線程式化干擾（Bit-line Program Disturbance，BPD），其發生原因在於程式化一元件後導致同一位元線之未程式化元件亦被程式化。
4. 位元線抹除干擾（Bit-line Erase Disturbance，BED），其發生原因在於程式化一元件後導致同一位元線之已程式化元件被抹除。
5. 讀取干擾（Read Disturbance，RD），其發生原因在於一元件被重複讀取後，由於浮動閘的漏電而無法保持在邏輯 0 之狀態。

裝  
訂  
線

## 五、發明說明 ( 7 )

6. 過度抹除 (Over Erase, OE)，其發生原因在於一元件被重複抹除；例如在抹除操作後，其臨界電壓因過低而導致該元件轉變為一空乏型電晶體。

一般記憶體之錯誤模式包含：

1. 固定值錯誤 (Stuck-At Fault, SAF)。
2. 轉換錯誤 (Transition Fault, TF)。
3. 固定開路錯誤 (Stuck-Open Fault, SOF)。
4. 位址解碼錯誤 (Address decoder Fault, AF)。
5. 狀態耦合錯誤 (State-Coupling Fault, SFst)。

本發明之演算法係以快閃記憶體元件陣列之兩對角線進行讀寫運算為主。第一對角線如圖1(a)至1(c)之實線所示，為一由左上至右下之 $-45^\circ$ 線。第二對角線如圖1(a)至1(c)之虛線所示，為一由左下至右上之 $+45^\circ$ 線，其中令m為列數，n為行數，且最低位址處位於左上角。本發明之演算法可適用於不同列數和行數之快閃記憶體元件陣列，例如圖1(a)代表該快閃記憶體元件陣列之列數等於行數的情形，圖1(b)代表該快閃記憶體元件陣列之列數小於行數的情形，圖1(c)代表該快閃記憶體元件陣列之列數大於行數的情形。在圖1(b)和圖1(c)之測試方法中可將該快閃記憶體虛擬為數個正方形，並以由上至下和左至右的方向執行。然為求方便敘述起見，以下之實施例係以列數小於等於行數之快閃記憶體元件陣列進行描述。

圖2係本發明之第一實施例之測試流程圖，包含如下之步驟，且各步驟之示意圖如圖3(a)至3(g)所示：

## 五、發明說明 ( 8 )

步驟21：抹除整個快閃記憶體元件陣列。其目的在使整個快閃記憶體元件陣列之值為邏輯1。

步驟22：如圖3(a)所示，又可分為三個子步驟：

- I. 讀取除第一對角線以外之元件。若其值不為邏輯1，代表出現SOF和CFst錯誤。
- II. 程式化除第一對角線以外之元件。其目的在偵測第一對角線之元件是否會出現WPD和BPD錯誤。
- III. 讀取除第一對角線以外之元件。若其值不為邏輯0，代表出現OE錯誤。

步驟23：如圖3(b)所示，又可分為三個子步驟：

- I. 讀取第一對角線之元件。若其值不為邏輯1，代表出現WPD和BPD錯誤。
- II. 程式化第一對角線之元件。其目的在偵測除第一對角線以外之元件是否會出現WED和BED錯誤。
- III. 讀取第一對角線之元件。若其值不為邏輯0，代表出現OE錯誤。

步驟24：如圖3(c)所示，讀取除第一對角線以外之元件。若其值不為邏輯0，代表出現WED和BED錯誤。

步驟25：如圖3(d)所示，程式化第二對角線之元件。其目的在偵測第一對角線之元件是否會出現WED和BED錯誤。

## 五、發明說明 (9)

步驟26：如圖3(e)所示，讀取第一對角線之元件。若其值不為邏輯0，代表出現WED和BED錯誤。

步驟27：抹除整個快閃記憶體元件陣列。其目的在使整個快閃記憶體元件陣列之值為邏輯1。

步驟28：如圖3(f)所示，又可分為三個子步驟：

I. 依相反於第一對角線之方向而讀取位於第一對角線之元件。其目的在偵測更多一般記憶體之錯誤。若其值不為邏輯1，代表出現SOF和CFst錯誤。

II. 依相反於第一對角線之方向而程式化位於第一對角線之元件。其目的在偵測除第一對角線以外之元件是否出現WPD和BPD錯誤。

III. 依相反於第一對角線之方向而讀取位於第一對角線之元件。其目的在偵測更多一般記憶體之錯誤。若其值不為邏輯0，代表出現CFst錯誤。

步驟29：如圖3(g)所示，讀取除第一對角線以外之元件。若其值不為邏輯1，代表出現WPD和BPD錯誤。

以下係依據本發明之測試方法之一較佳實施例之演算法：

### 1. Flash Erase;

// 令*i*為該快閃記憶體元件陣列之列位址註標，*j*為該快閃記憶體元件陣列之行位址註標，*x*為*m*和*n*中之較小者

## 五、發明說明 (10 )

2. for  $j=0$  to  $(n-1)$  // 除第一對角線以外之元件

```

for i=0 to (m-1)
if (((j-i) % m)!=0)
{ Read1 (i,j);
Program (i,j);
Read0 (i,j);}

```

3. for  $j=0$  to  $(n-1)$  // 第一對角線之元件

```

for i=0 to (m-1)
if (((j-i) % m)=0)
{ Read1 (i,j);
Program (i,j);
Read0 (i,j);}

```

4. for  $j=0$  to  $(n-1)$  // 除第一對角線以外之元件

```

for i=0 to (m-1)
if (((j-i) % m)!=0) Read0 (i,j);

```

5. for  $j=0$  to  $(n-1)$  // 第二對角線之元件

```

for i=(m-1) to 0
if (((j-((m-1)-i)) % m)=0) Program (i,j);

```

6. for  $j=0$  to  $(n-1)$  // 第一對角線之元件

```

for i=0 to (m-1)
if (((j-i) % m)=0) Read0 (i,j);

```

7. Flash Erase;

8. for  $j=(n-1)$  to 0 // 第一對角線之元件

```

for i=(m-1) to 0

```

## 五、發明說明(11)

```

if(((j-i) % m)=0)
{ Read1 (i,j);
Program (i,j);
Read0 (i,j); }
9. for j=0 to (n-1) // 除第一對角線以外之元件
for i=0 to (m-1)
if (((j-i) % m)!=0) Read1 (i,j);

```

本發明之演算法可以用以下之簡潔型式表示：

$$\{(E); \uparrow_{D_1}(R1, P, R0); \uparrow_{D_1}(R1, P, R0); \uparrow_{D_1}(R0); \downarrow_{D_2}(P); \uparrow_{D_1}(R0); (E); \downarrow_{D_1}(R1, P, R0); \uparrow_{D_1}(R1)\}$$

其中符號↑代表元件位址漸增，符號↓代表元件位址漸減， $D_1$ 代表第一對角線之元件， $D_2$ 代表第二對角線之元件， $!D_1$ 代表除第一對角線以外之元件， $!D_2$ 代表除第二對角線以外之元件； $R1$ 代表讀取元件，若其值不為1則代表錯誤被偵測出； $P$ 代表程式化元件； $R0$ 代表讀取元件，若其值不為0則代表錯誤被偵測出。

值得注意的是，上述實施例之錯誤模式及演算法係以NOR閘型態的堆疊閘型快閃記憶體為例，然實際應用時並無特定之限制。另外，對RD錯誤而言，由於讀取一元件之操作電壓係等同於WED錯誤時受影響元件之操作電壓，因此RD錯誤之情形可以藉由WED錯誤之檢測方式而同時被檢測出來。

本發明之上述實施例之演算法對AF和CFst錯誤之錯誤涵蓋率並未達到100%。一種可行的方式係以如下的方式

## 五、發明說明 (12)

修改步驟29：

步驟29：又可分為三個子步驟：

- I. 依高位址至低位址之方向而讀取除第一對角線以外之元件。若其值不為邏輯1，代表出現SOF和CFst錯誤。
- II. 依高位址至低位址之方向而程式化除第一對角線以外之元件。
- III. 依高位址至低位址之方向而讀取除第一對角線以外之元件。若其值不為邏輯0，代表出現OE錯誤。

修正後之步驟29之一可行之演算法如下：

```

10. for j=(n-1) to 0 // (R1, P, R0) 所有元件
    for i=(m-1) to 0 // 除第一對角線以外之元件
        if (((j-i) % m)!=0)
            { Read1 (i,j);
              Program (i,j);
              Read0 (i,j); }

```

修正後之演算法可以用以下之簡潔型式表示：

$$\{(E); \overset{D_1}{\uparrow}(R1, P, R0); \overset{D_1}{\uparrow}(R1, P, R0); \overset{D_1}{\uparrow}(R0); \overset{D_2}{\downarrow}(P); \overset{D_1}{\uparrow}(R0); (E); \overset{D_1}{\downarrow}(R1, P, R0); \overset{D_1}{\uparrow}(R1, P, R0)\}$$

修正後之演算法（以下稱為對角線延伸型）之測試時間僅微小地增加，而測試長度則變為 $2e + (2mn + max(m, n))p + 5mn r$ ，其中參數 $e$ 代表抹除時間，參數 $p$ 代表程式化時間，參數 $r$ 代表讀取時間。對2MB

## 五、發明說明 (13)

的快閃記憶體而言，測試時間大約為 6.711 秒，大約相等於 March-like 演算法。然若僅考量快閃記憶體之干擾錯誤模式之偵錯，則本發明之第一實施例之演算法較為妥適，因其具有較短之測試時間，可大幅縮小測試之成本。

圖 4(a) 至 4(i) 係本發明之第一實施例之另一分解表示圖（以下稱為對角線型）。值得注意的是，圖 4(g) 至 4(i) 之步驟相對於圖 4(a) 至 4(f) 之步驟係屬於較獨立之流程，因此亦可將圖 4(g) 至 4(i) 之步驟設置於圖 4(a) 至 4(f) 之步驟之前。重組後之流程如圖 5(a) 至 5(i) 所示，即本發明之第二實施例。

圖 6(a) 至 6(i) 係本發明之第三實施例之測試步驟分解圖，其係改良圖 4(a) 至 4(i) 之第一實施例之步驟順序。該第三實施例將圖 4(e) 和 4(f) 之步驟移至整個流程之最後（以下稱為對角線修正型）。

值得注意的是，圖 6(e) 至 6(i) 之步驟相對於圖 6(a) 至 6(d) 之步驟係屬於獨立之流程，因此亦可將圖 6(e) 至 6(i) 之步驟設置於圖 6(a) 至 6(d) 之步驟之前。重組後之流程如圖 7(a) 至 7(i) 所示，即本發明之第四實施例。

圖 8(a) 至 8(i) 係本發明之第五實施例之測試步驟分解圖，其係使用對角線延伸型之演算法（以下稱為對角線延伸型），因此可得到 100% 的錯誤涵蓋率。

圖 9(a) 至 9(h) 係本發明之第六實施例之測試步驟分解圖，其係改良圖 8(a) 至 8(i) 之第五實施例之步驟順序。該第六實施例將步驟 8(f) 調至測試流程之最後一步驟，且省

裝  
訂  
線

## 五、發明說明 (14)

略圖 8(e)之步驟（以下稱為對角線延伸修正型）。

類似以上之情形，圖 9(e)至 9(h)之步驟相對於圖 9(a)至 9(d)之步驟係屬於獨立之流程，因此亦可將圖 9(e)至 9(h)之步驟設置於圖 9(a)至 9(d)之步驟之前。重組後之流程如圖 10(a)至 10(h)之本發明之第七實施例所示。

圖 11(a)至 11(h)係本發明之第八實施例之測試步驟分解圖，其係改良圖 8(a)至 8(i)之第五實施例之步驟順序。該第八實施例將步驟 8(g)至 8(i)調至測試流程之起始。

表 1 係顯示經由 RAMSES-FT 偵錯模擬器（請見於 K.L. Cheng, J.C. Yeh, C.W. Wang, C.T. Huang, and C.W. Wu, "RAMSES-FT : A Fault Simulator for Flash Memory Testing and Diagnostics", Proc. IEEE VLSI Test Symp., Monterey, California, Apr. 2002 之論文內）比較本發明和習知技藝之測試時間和測試長度後之特性。表 2 則顯示本發明和習知技藝之錯誤涵蓋率之比較結果。上述實驗係使用 2 MB (256K×8) 快閃記憶體為例，其抹除時間為 200 ms，位元程式化時間為 12 μs，位元讀取時間為 10 ns。此外，上述實驗僅使用位元導向演算法。

表 1

演算法	測試長度 (Test Length)	複雜度 Complexity ( $N = m \times n$ )			測試時間(秒) 以 100 MHz 為例
		抹除	程式化	讀取	
EF	$2e + (mn + 2m + n - 2)p + (2mn + m + n - 3)r$	2	$1N + 3\sqrt{N}$	$2N + 2\sqrt{N}$	3.569

## 五、發明說明 (15)

Flash March	$2e + (2mn)p + (4mn)r$	2	$2N$	$4N$	6.702
March-FT	$2e + (2mn)p + (6mn)r$	2	$2N$	$6N$	6.707
本發明之對角線型	$2e + (mn + 2\max(m, n))p + (4mn + \max(m, n))r$	2	$1N + 2\sqrt{N}$	$4N + \sqrt{N}$	3.569
本發明之對角線修正型	$2e + (mn + 2\max(m, n))p + (4mn + 3\max(m, n))r$	2	$1N + 2\sqrt{N}$	$4N + 3\sqrt{N}$	3.569
本發明之對角線延伸型	$2e + (2mn + \max(m, n))p + (5mn)r$	2	$2N + \sqrt{N}$	$5N$	6.711
本發明之對角線延伸修正型	$2e + (2mn)p + (5mn)r$	2	$2N$	$5N$	6.707

表 2

	WPD	WED	BPD	BED	OE	RD
EF	100%	100%	0%	100%	100%	100%
Flash March	100%	100%	100%	100%	100%	100%
March-FT	100%	100%	100%	100%	100%	100%
對角線型	100%	100%	100%	100%	100%	100%
對角線修正型	100%	100%	100%	100%	100%	100%
對角線延伸型	100%	100%	100%	100%	100%	100%
對角線延伸修正型	100%	100%	100%	100%	100%	100%

裝訂線

表 2 (續)

	SAF	TF	SOF	AF	CFst	總錯誤涵蓋率
EF	100%	87.5%	12.5%	44.5%	50%	72.23%

## 五、發明說明(16)

Flash March	100%	100%	50%	100%	75%	93.18%
March-FT	100%	100%	100%	100%	100%	100%
對角線型	100%	100%	100%	72.5%	86.25%	96.25%
對角線修正型	100%	100%	100%	81.6%	89.15%	97.34%
對角線延伸型	100%	100%	100%	100%	100%	100%
對角線延伸修正型	100%	100%	100%	100%	100%	100%

經由表1和表2之顯示結果，足可證明本發明之測試長度較短，且對干擾錯誤及一般記憶體錯誤具有較佳之錯誤涵蓋率。此外，因本發明之測試方法具有規則性及對稱性，亦使依據本發明之對角線測試方法所製成之內建自我測試電路更易製作。

本發明之技術內容及技術特點已揭示如上，然而熟悉本項技術之人士仍可能基於本發明之教示及揭示而作種種不背離本發明精神之替換及修飾。因此，本發明之保護範圍應不限於實施例所揭示者，而應包括各種不背離本發明之替換及修飾，並為以下之申請專利範圍所涵蓋。

裝  
訂  
線

## 六、申請專利範圍

(請先閱讀背面之注意事項再填寫本頁)

1. 一種快閃記憶體之對角線測試方法，包含：
  - (a) 抹除該快閃記憶體元件陣列；
  - (b) 程式化該快閃記憶體元件陣列之除第一對角線以外之元件；
  - (c) 讀取該快閃記憶體元件陣列之第一對角線之元件；
  - (d) 程式化該快閃記憶體元件陣列之第一對角線之元件；及
  - (e) 讀取該快閃記憶體元件陣列之除第一對角線以外之元件。
2. 如申請專利範圍第1項之快閃記憶體之對角線測試方法，其另包含以下步驟：
  - (f) 程式化該快閃記憶體元件陣列之第二對角線之元件；及
  - (g) 讀取該快閃記憶體元件陣列之第一對角線之元件。
3. 如申請專利範圍第1項之快閃記憶體之對角線測試方法，其中該第一對角線係指該快閃記憶體元件陣列之左上至右下之-45度方向對角線，且該第二對角線係指該快閃記憶體元件陣列之左下至右上之+45度方向對角線。
4. 如申請專利範圍第1項之快閃記憶體之對角線測試方法，其中在步驟(a)之後另包含讀取該快閃記憶體元件陣列之除第一對角線以外之元件之步驟。
5. 如申請專利範圍第1項之快閃記憶體之對角線測試方法，其中在步驟(b)之後另包含讀取該快閃記憶體元件陣列之除第一對角線以外之元件之步驟。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

## 六、申請專利範圍

6. 如申請專利範圍第1項之快閃記憶體之對角線測試方法，其中在步驟(d)之後另包含讀取該快閃記憶體元件陣列之第一對角線之元件之步驟。
7. 如申請專利範圍第1項之快閃記憶體之對角線測試方法，其另包含以下步驟：
  - (h)抹除該快閃記憶體元件陣列；
  - (i)依相反於第一對角線之方向而程式化位於該快閃記憶體元件陣列之第一對角線之元件；及
  - (j)讀取該快閃記憶體元件陣列之除第一對角線以外之元件。
8. 如申請專利範圍第7項之快閃記憶體之對角線測試方法，其中在步驟(h)之後另包含依相反於第一對角線之方向而讀取位於該快閃記憶體元件陣列之第一對角線之元件之步驟。
9. 如申請專利範圍第7項之快閃記憶體之對角線測試方法，其中在步驟(i)之後另包含依相反於第一對角線之方向而讀取位於該快閃記憶體元件陣列之第一對角線之元件之步驟。
10. 如申請專利範圍第1項之快閃記憶體之對角線測試方法，其另包含下列步驟：
  - (h)抹除該快閃記憶體元件陣列；
  - (i)依相反於第一對角線之方向而程式化位於該快閃記憶體元件陣列之第一對角線之元件；
  - (j)依高位址至低位址之方向而程式化位於該快閃記憶

(請先閱讀背面之注意事項再填寫本頁)

裝訂線

## 六、申請專利範圍

體元件陣列之除第一對角線以外之元件；及

(k) 依高位址至低位址之方向而讀取位於該快閃記憶體元件陣列之除第一對角線以外之元件。

11. 如申請專利範圍第10項之快閃記憶體之對角線測試方法，其另包含讀取該快閃記憶體元件陣列之第一對角線之元件之步驟。

12. 如申請專利範圍第1項之快閃記憶體之對角線測試方法，其係將該快閃記憶體元件陣列虛擬為數個正方形，並以由上至下和左至右的方向執行。

13. 一種快閃記憶體之對角線測試方法，包含：

抹除該快閃記憶體元件陣列；

依相反於第一對角線之方向而程式化位於該快閃記憶體元件陣列之第一對角線之元件；

依高位址至低位址之方向而程式化位於該快閃記憶體元件陣列之除第一對角線以外之元件；及

依高位址至低位址之方向而讀取位於該快閃記憶體元件陣列之除第一對角線以外之元件。

14. 如申請專利範圍第13項之快閃記憶體之對角線測試方法，其另包含讀取該快閃記憶體元件陣列之第一對角線之元件之步驟。

15. 如申請專利範圍第13項之快閃記憶體之對角線測試方法，其另包含下列步驟：

抹除該快閃記憶體元件陣列；

程式化該快閃記憶體元件陣列之除第一對角線以外

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

線

## 六、申請專利範圍

之元件；

讀取該快閃記憶體元件陣列之第一對角線之元件；

程式化該快閃記憶體元件陣列之第一對角線之元件；及

讀取該快閃記憶體元件陣列之除第一對角線以外之元件。

16. 如申請專利範圍第13項之快閃記憶體之對角線測試方法，其另包含下列步驟：

程式化該快閃記憶體元件陣列之第二對角線之元件；及

讀取該快閃記憶體元件陣列之第一對角線之元件。

17. 如申請專利範圍第13項之快閃記憶體之對角線測試方法，其中該第一對角線係指該快閃記憶體元件陣列之左上至右下之-45度方向對角線，且該第二對角線係指該快閃記憶體元件陣列之左下至右上之+45度方向對角線。

18. 一種快閃記憶體之對角線測試方法，包含：

抹除該快閃記憶體元件陣列；

依相反於第一對角線之方向而程式化位於該快閃記憶體元件陣列之第一對角線之元件；及

讀取該快閃記憶體元件陣列之除第一對角線以外之元件。

19. 如申請專利範圍第18項之快閃記憶體之對角線測試方法，其另包含下列步驟：

## 六、申請專利範圍

(請先閱讀背面之注意事項再填寫本頁)

裝-----線-----訂-----

抹除該快閃記憶體元件陣列；

程式化該快閃記憶體元件陣列之除第一對角線以外之元件；

讀取該快閃記憶體元件陣列之第一對角線之元件；

程式化該快閃記憶體元件陣列之第一對角線之元件；及

讀取該快閃記憶體元件陣列之除第一對角線以外之元件。

20. 如申請專利範圍第18項之快閃記憶體之對角線測試方法，其另包含下列步驟：

程式化該快閃記憶體元件陣列之第二對角線之元件；及

讀取該快閃記憶體元件陣列之第一對角線之元件。

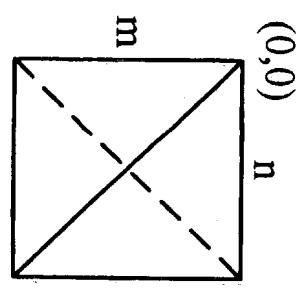


圖 1(a)

—— 第一對角線1(D1)  
----- 第二對角線2(D2)

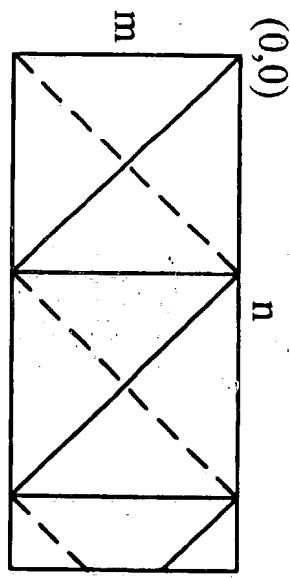


圖 1(b)

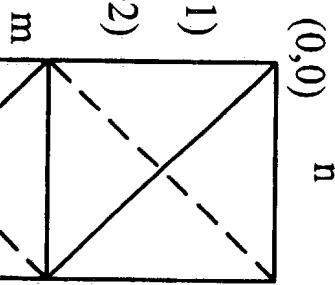


圖 1(c)

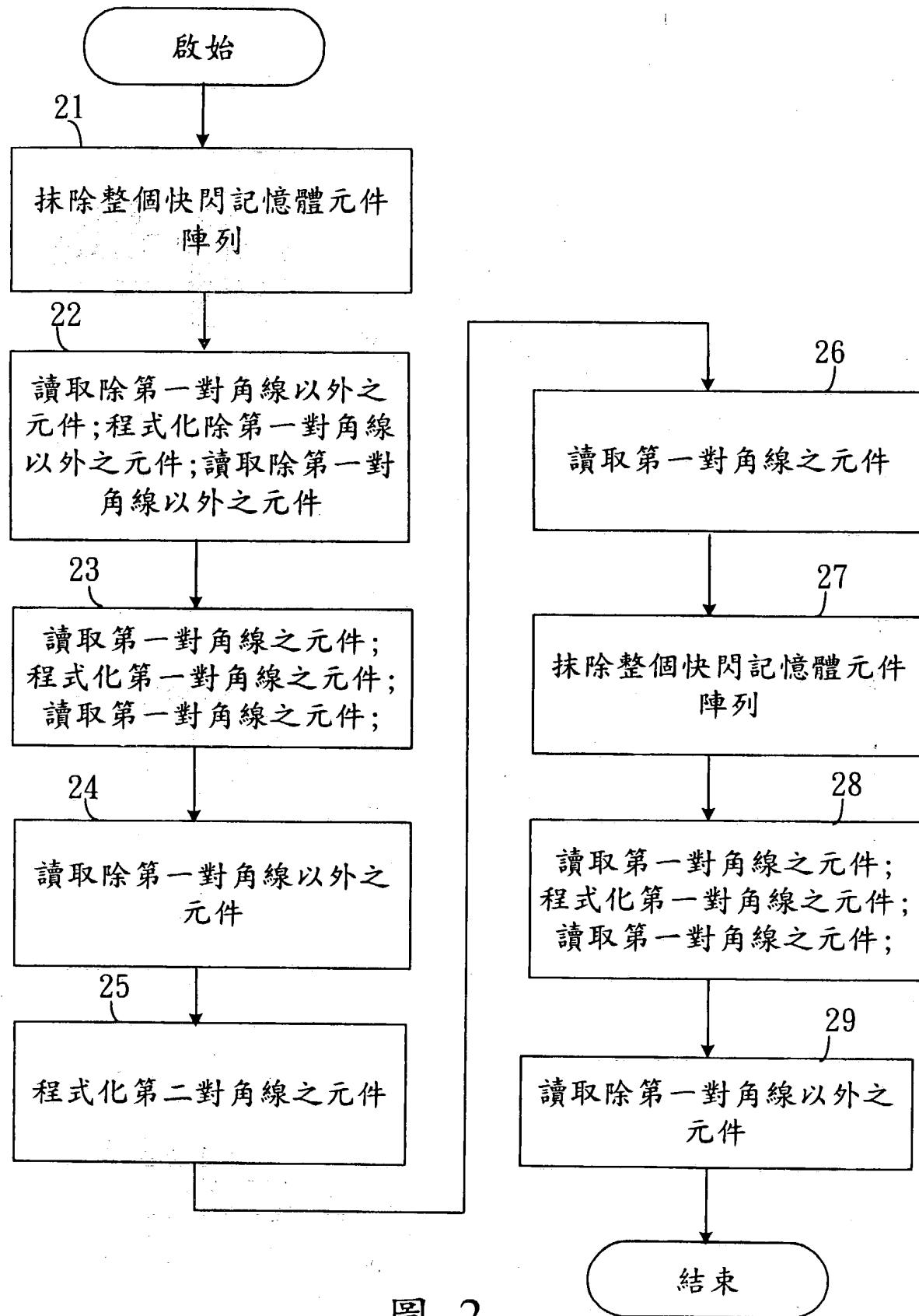


圖 2

*Step 22 :  $\uparrow_{D_1}(R1, P, R0)$*

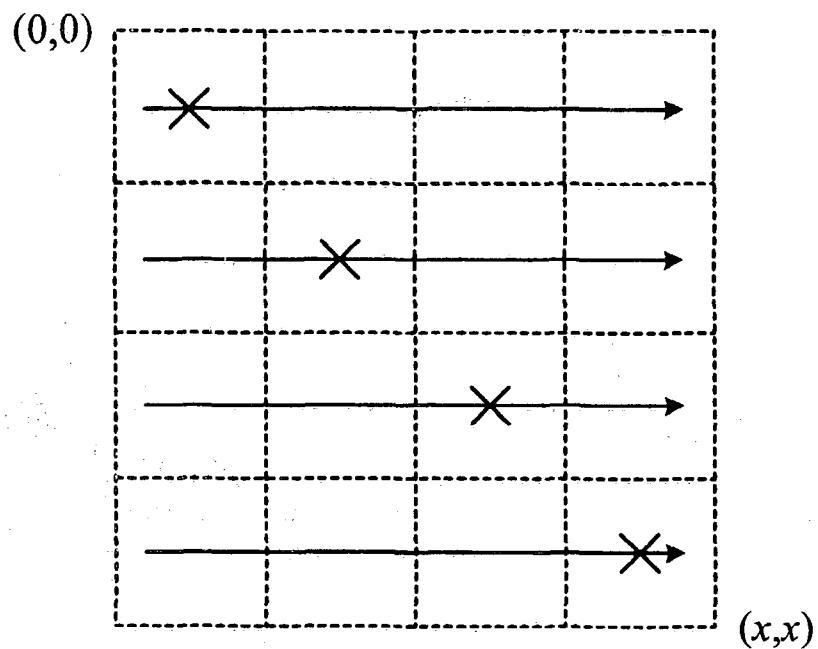


圖 3(a)

*Step 23 :  $\uparrow_{D_1}(R1, P, R0)$*

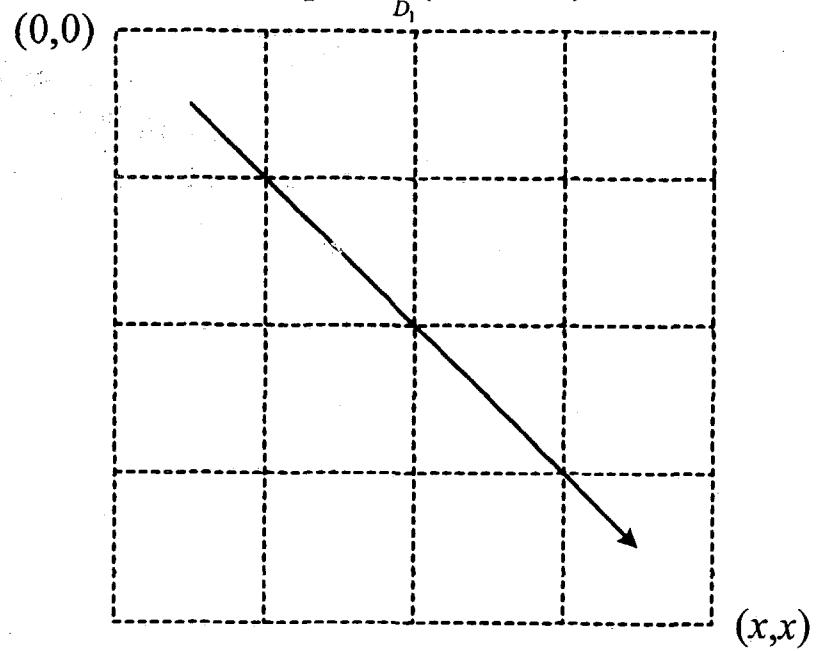
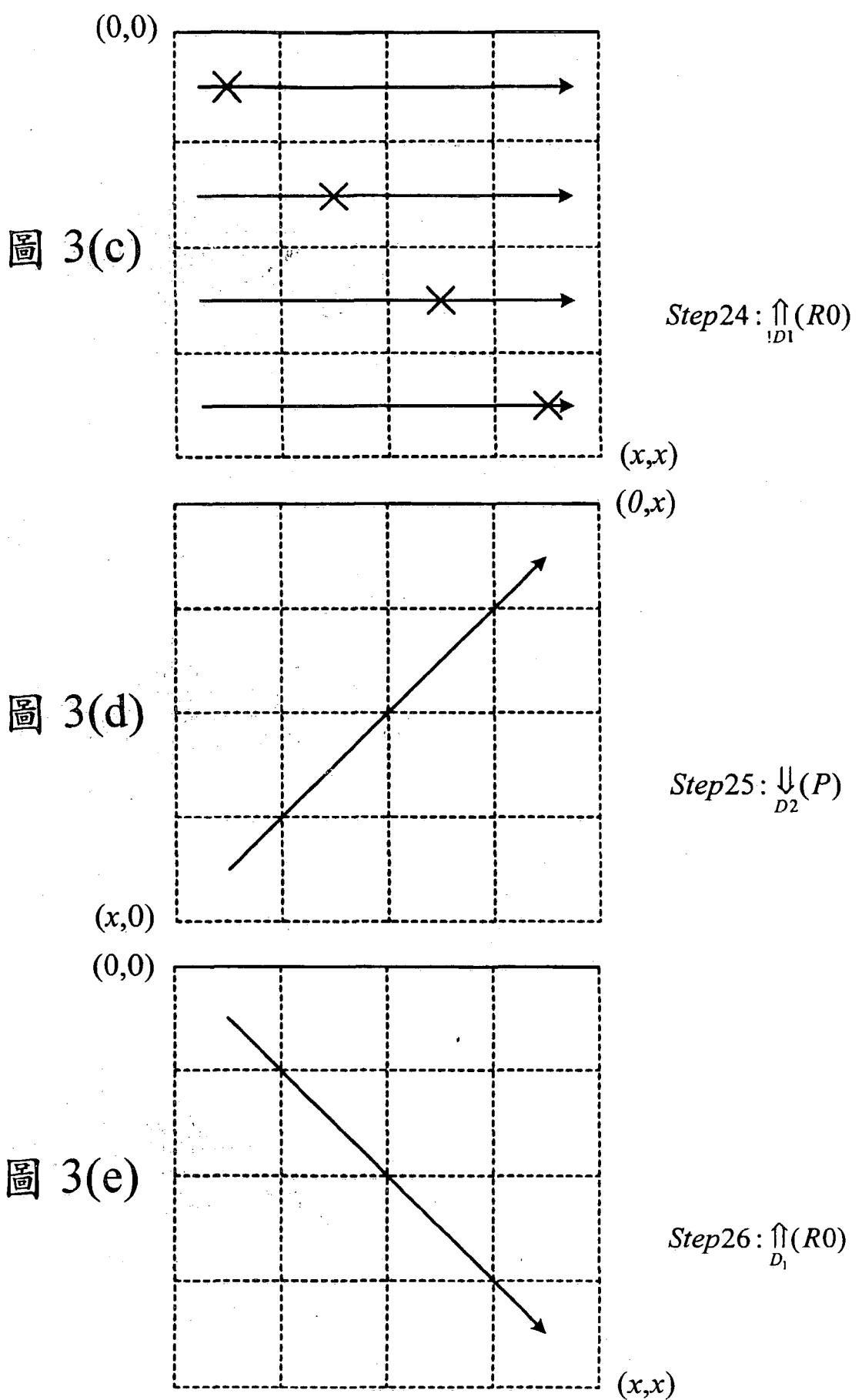


圖 3(b)



*Step 8:*  $\downarrow_{D1}(R1, P, R0)$

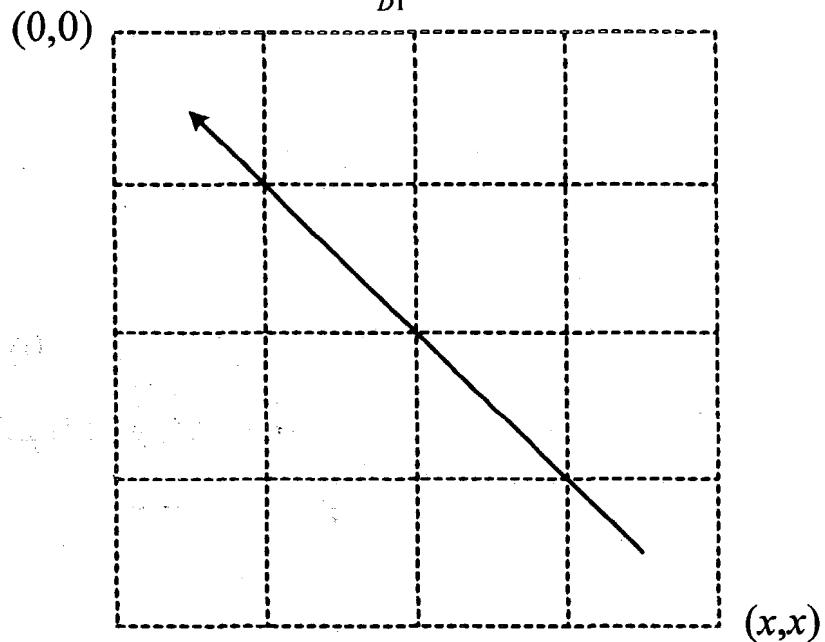


圖 3(f)

*Step9:*  $\underset{!D1}{\uparrow}(R1)$

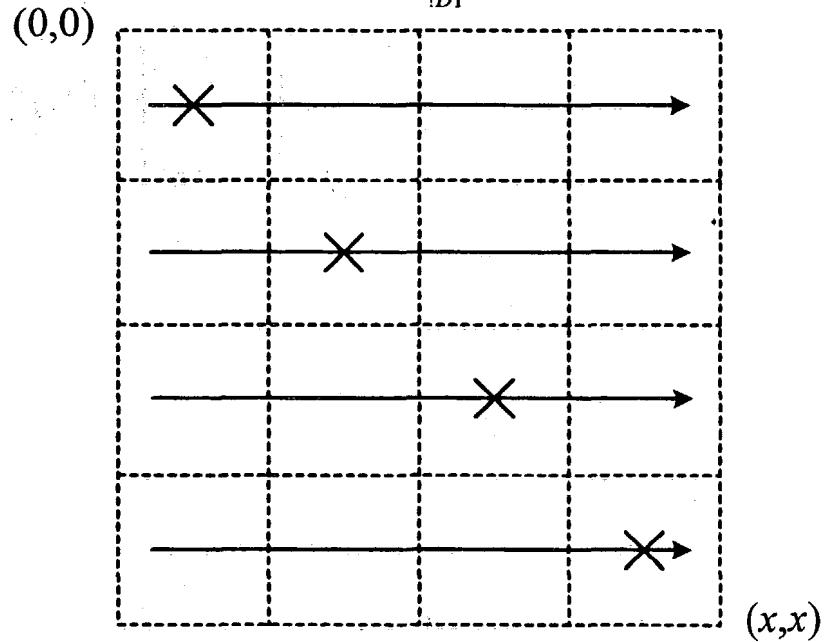


圖 3(g)

圖 4(f)

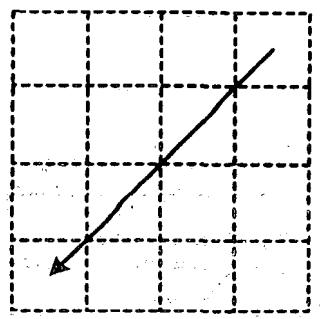


圖 4(g)

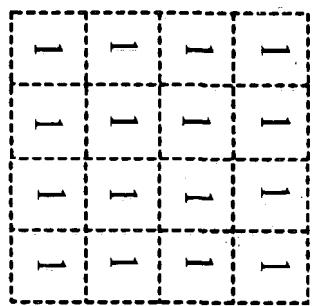


圖 4(h)

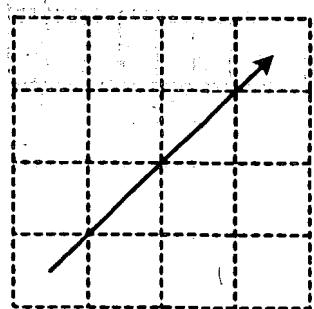


圖 4(i)

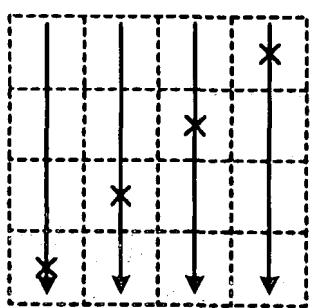


圖 4(a)

圖 4(b)

圖 4(c)

圖 4(d)

圖 4(e)

抹除			
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

$\uparrow(R1, P, R0)$			
*			
	*		
		*	
			*
			*

$\uparrow(R1, P, R0)$			
*			
	*		
		*	
			*
			*

$\uparrow(R1)$			
*			
	*		
		*	
			*
			*

$\Downarrow(P)$			

圖 5(f)

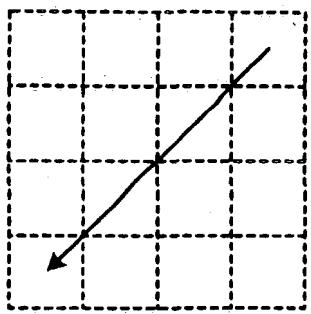


圖 5(g)

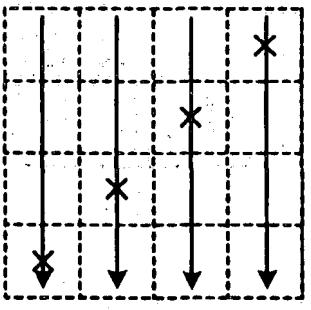


圖 5(h)

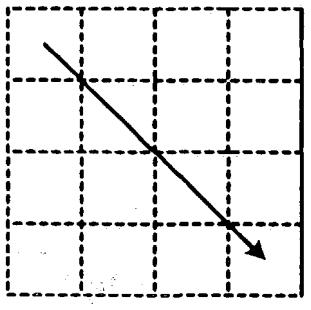


圖 5(i)

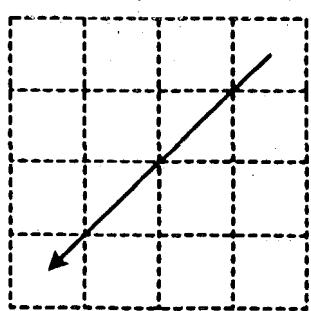


圖 5(a)

$\uparrow(R1, P, R0)$   
 $D_1$

$\uparrow(R0)$   
 $D_1$

$\downarrow(P)$   
 $D_2$

$\uparrow(R0)$   
 $D_1$

圖 5(b)

圖 5(c)

圖 5(d)

圖 5(e)

抹除			
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

$\downarrow(R1, P, R0)$ $D_1$			
*			
*			
*			
*			

$\uparrow(R1)$ $D_1$			
*			
*			
*			
*			

$\uparrow(R1, P, R0)$ $D_1$			
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

$\uparrow(R1, P, R0)$ $D_1$			
*			
*			
*			
*			

圖 6(f)

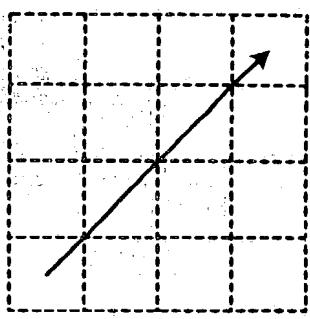


圖 6(g)

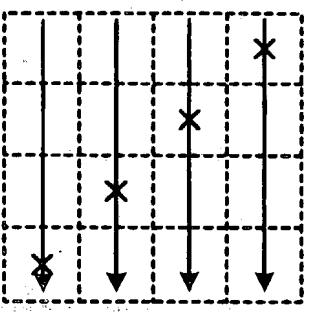


圖 6(h)

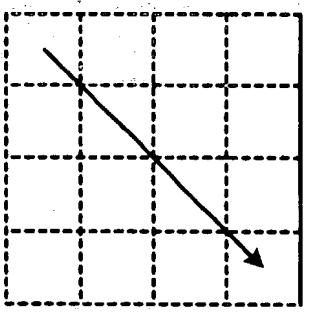


圖 6(i)

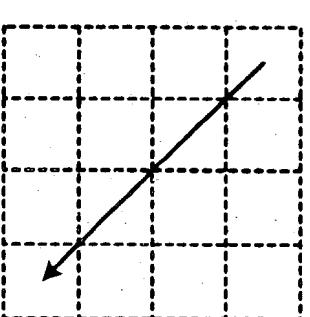


圖 6(a)

$\Downarrow_{D_1}(R1, P, R0)$

圖 6(b)

$\Downarrow_{D_1}(R1)$

圖 6(c)

$\Downarrow_{D_2}(R1, P, R0)$

圖 6(d)

$\Downarrow_{D_1}(R0)$

圖 6(e)

抹除

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

$\Downarrow_{D_1}(R1, P, R0)$

$\Downarrow_{D_1}(R1, P, R0)$

$\Downarrow_{D_1}(R0)$

抹除

*			
*			
*			
*			
*			


*			
*			
*			
*			
*			

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

圖 7(f)

圖 7(g)

圖 7(h)

圖 7(i)

抹除			
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

抹除			
*			
*			
*			
*			

抹除			

抹除			
*			
*			
*			
*			

圖 7(a)

圖 7(b)

圖 7(c)

圖 7(d)

圖 7(e)

抹除			
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

抹除			
*			
*			
*			
*			

抹除			

抹除			
*			
*			
*			
*			

抹除			

↑(R1, P, R0)  
 $\uparrow_{D_1}$

↓(R1, P, R0)  
 $\downarrow_{D_1}$

↑(R1)  
 $\uparrow_{D_1}$

↓(R1, P, R0)  
 $\downarrow_{D_2}$

↑(R0)  
 $\uparrow_{D_1}$

↓(R0)  
 $\downarrow_{D_1}$

圖 8(f)

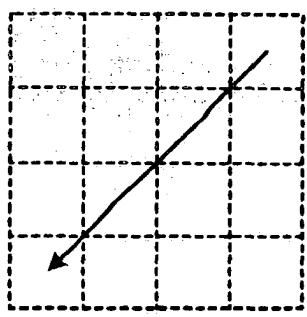


圖 8(g)

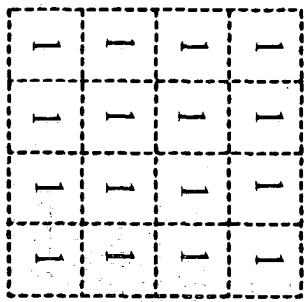


圖 8(h)

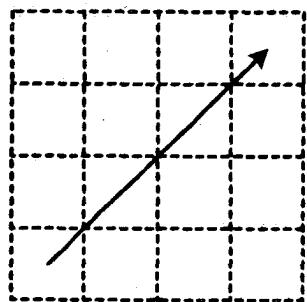


圖 8(i)

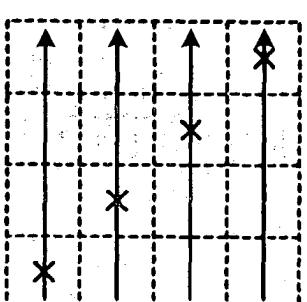


圖 8(a)

$\overset{\uparrow}{\uparrow}(R0)$

抹除			
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

圖 8(b)

抹除

$\overset{\uparrow}{\uparrow}(R1, P, R0)$			
*			
	*		
		*	
			*

圖 8(c)

$\overset{\downarrow}{\uparrow}(R1, P, R0)$

$\overset{\uparrow}{\uparrow}(R1, P, R0)$			
*			
	*		
		*	
			*

圖 8(d)

$\overset{\downarrow}{\downarrow}(R1, P, R0)$

$\overset{\downarrow}{\downarrow}(R1, P, R0)$			
↔			
	*		
		*	
			*

圖 8(e)

$\overset{\downarrow}{\downarrow}(R1, P, R0)$

$\overset{\downarrow}{\downarrow}(R1, P, R0)$			

抹除

$\overset{\uparrow}{\uparrow}(R1, P, R0)$

$\overset{\uparrow}{\uparrow}(R1, P, R0)$

$\overset{\downarrow}{\downarrow}(R1, P, R0)$

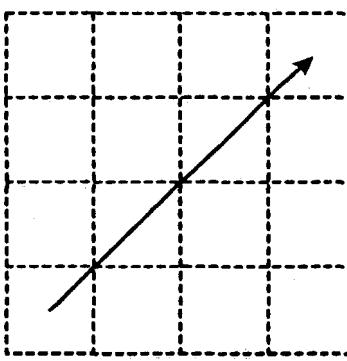
$\overset{\downarrow}{\downarrow}(P)$

圖 9(e)

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

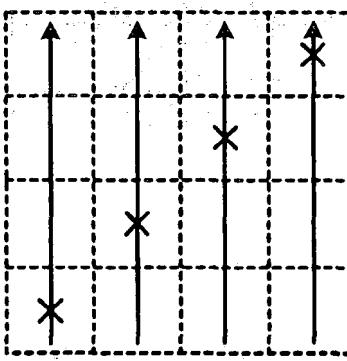
抹除

圖 9(f)



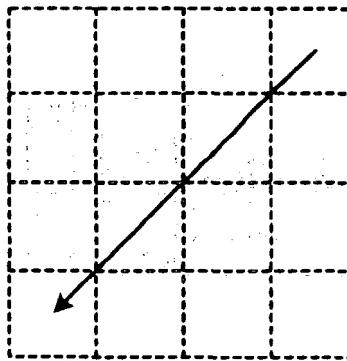
$\Downarrow_{D_1}(R1, P, R0)$

圖 9(g)



$\Downarrow_{D_1}(R1, P, R0)$

圖 9(h)



$\Downarrow_{D_1}(R0)$

圖 9(a)

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

抹除

$\Uparrow_{D_1}(R1, P, R0)$

*			
	*		
		*	
			*
			*

圖 9(b)

*			
	*		
		*	
			*
			*

圖 9(c)

*			
	*		
		*	
			*
			*

圖 9(d)

$R = S$

圖 10(e)

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

圖 10(f)

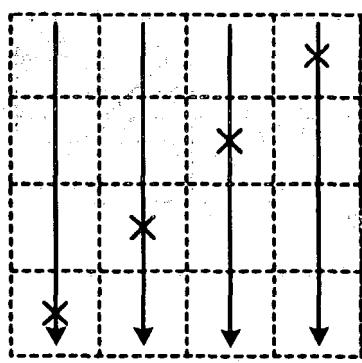


圖 10(g)

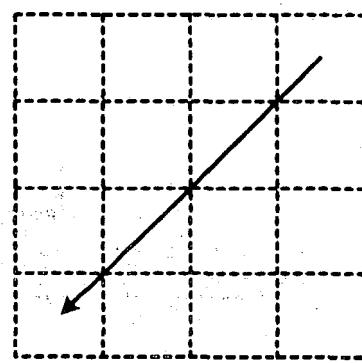


圖 10(h)

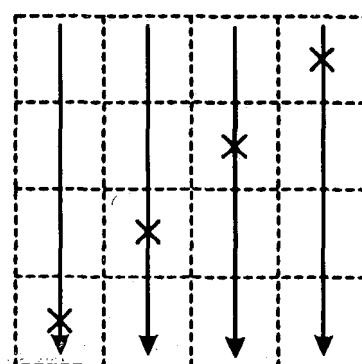


圖 10(a)

抹除

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

圖 10(b)

$\Downarrow_{D_1}(R1, P, R0)$

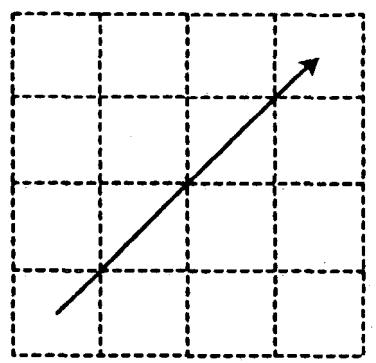


圖 10(c)

$\Downarrow_{D_1}(R1, P, R0)$

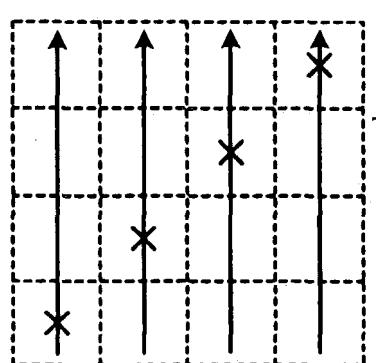
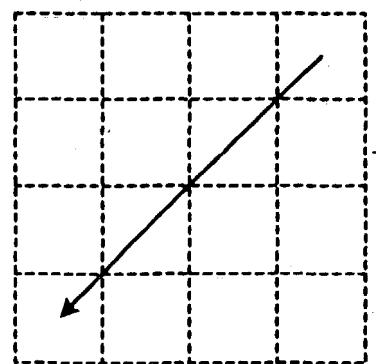


圖 10(d)

$\Downarrow_{D_1}(R0)$





J - 6

卷一

抹除

$$\Downarrow_{D_1}(R1, P, R0)$$

$$\Downarrow_{D_1}(R1, P, R0)$$

抹除

$$\Downarrow_{D_1}(R1, P, R0)$$

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1


*			
	*		
		*	
			*


*			
	*		
		*	
			*

圖 11(a)

圖 11(b)

圖 11(c)

圖 11(d)

圖 11(e)

$$\Uparrow_{D_1}(R1, P, R0)$$

$$\Uparrow_{D_1}(R0)$$

$$\Downarrow_{D_2}(P)$$

$$\Uparrow_{D_1}(R0)$$


*			
	*		
		*	
			*


圖 11(f)

圖 11(g)

圖 11(h)

圖 11(i)